APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/635,815	08/07/2003	Jay D. Knitter	200209059-1	9770

22879 7590 05/14/2008
HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER
BIAGINI, CHRISTOPHER D

ART UNIT	PAPER NUMBER
2142	

NOTIFICATION DATE	DELIVERY MODE
05/14/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
mkraft@hp.com
ipa.mail@hp.com

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/635,815
Filing Date: August 07, 2003
Appellant(s): KNITTER, JAY D.

William T. Ellis and Walter K. Robinson
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed April 17, 2008 appealing from the Office action

mailed September 26, 2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,457,066   Mein et al.   9-2002

Apache Axis Home Page, Apache Software Foundation, accessed 4/11/2007:
<http://ws.apache.org/axis.>. Pages 1-2.

White, Jim. "Re: Finding the caller in java". USENET Post. April 19, 2000. Archived by Google at <http://groups.google.com/group/comp.lang.java.programmer/msg/b24afb100ccdfe8b> Page 1.

Winer, Dave. "Scripting News: 9/27/2002": <http://www.scripting.com/2002/09/27.html>. September 27, 2002.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

35 USC § 112, First Paragraph

Claims 1-22 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with

the enablement requirement. The claim(s) contains subject matter which was not described in

the specification in such a way as to enable one skilled in the art to which it pertains, or with

which it is most nearly connected, to make and/or use the invention.

The test of enablement is whether one reasonably skilled in the art could make or use the

invention from the disclosures in the patent coupled with information known in the art without

undue experimentation. *United States v. Telectronics, Inc.*, 857 F.2d 778, 785, 8 USPQ2d 1217,

1223 (Fed. Cir. 1988). The factors to be considered when determining whether there is sufficient

evidence to support a determination that a disclosure does not satisfy the enablement requirement

and whether any necessary experimentation is "undue" include, but are not limited to: (a) the

breadth of the claims; (b) the nature of the invention; (c) the state of the prior art; (d) the level of

one of ordinary skill; (e) the level of predictability in the art; (f) the amount of direction provided

by the inventor; (g) the existence of working examples; and (h) the quantity of experimentation

needed to make or use the invention based on the content of the disclosure. *In re Wands,* 858

F.2d 731, 737, 8 USPQ2d 1400, 1404 (Fed. Cir. 1988).

As to the breadth of the claims, the currently pending claims in the instant application are broad. Essentially all implementations for using an intermediary to examine an execution stack to identify the caller of a network service fall within the scope of the claims. Analysis of this factor in light of all the evidence of record therefore suggests that the amount of experimentation required to make and use the invention is undue.

As to the nature of the invention, there is no relevant evidence of probative value in the record. Analysis of this factor in light of all the evidence of record therefore does not suggest whether the amount of experimentation required to make and use the invention is undue or not undue.

As to the state of the prior art, the very features that Appellant argues (see Remarks, July 16, 2007) distinguish the claims from the prior art are those that are not adequately described. A search of the prior art has not revealed a system as claimed for using an intermediary to examine an execution stack to identify the caller of a network service. Analysis of this factor in light of all the evidence of record therefore suggests that the amount of experimentation required to make and use the invention is undue.

As to the level of skill in the art, there is no evidence in the record as to the level of skill in the art. Analysis of this factor in light of all the evidence of record therefore does not suggest

whether the amount of experimentation required to make and use the invention is undue or not undue.

As to the level of predictability in the art, the computer arts are generally considered predictable. Analysis of this factor in light of all the evidence of record therefore suggests that the amount of experimentation required to make and use the invention is not undue.

As to the amount of direction provided by the inventor, Appellants's arguments (see Remarks, July 16, 2007) indicate that the distinguishing feature of the claimed invention is the use of a "comparison algorithm" on a client server to identify "an object on the client computer that is invoking the object on [a] data server." However, the specification does not adequately describe how this operation occurs. It is unclear whether the client server examines its own execution stack or the execution stack of the client computer. Examining its own execution stack would be of little benefit. Since the client computer and client server are completely separate machines, as illustrated in Fig. 1, and the execution stack of a system contains only those objects which exist locally in memory, the execution stack of the client server would contain no information regarding which object on the client computer was invoking the object. If the client server is intended to examine the execution stack of the client computer, it is unclear how this would occur, as there is no direction in the specification as to how the client server is able to examine the execution stack of a remote system.

The specification describes exemplary code for implementing the claimed algorithm, but this code would not be functional in the claimed embodiment. Specifically, and in stark contrast

to the claimed embodiment and the embodiment illustrated in Fig. 1, it is noted that *the client executes on the same machine as the comparison algorithm*. Entry into the program occurs at the `main` method on line 2. Subsequently, the program creates a `Client2` object and invokes its `sendSOAPMessage` method. The `sendSOAPMessage` method in turn creates a `Client3` object and invokes its `findSourceOfSOAPMessage` method. The `findSourceOfSOAPMessage` method then creates an `Algorithm` object and invokes its `findSourceOfSOAPMEssageCall` method. The execution stack is accessed on line 13 of the `Algorithm` class, but this execution stack is merely the local execution stack shared with the initial `Client` object. In other words, such an algorithm would not be functional in an arrangement where the client and client server are not local objects running on a single machine, but computers separated by a network link.

Furthermore, although Appellants arguments and amended claim limitations (see response filed July 16, 2007) indicate that the algorithm identifies an *object* on the client computer that is invoking the object on the data server, absolutely no provision for doing so appears in the specification. Even assuming, *arguendo*, that the described algorithm is functional in the claimed embodiment, the algorithm only identifies the *class* that invokes the object. In object-oriented programming, classes are essentially "blueprints" for creating objects in memory. A class may be instantiated into multiple different objects, each of which share the class's name. Therefore, the class name returned by the algorithm described on pages 5 and 6 of the specification is insufficient to "identify…an object on the client computer." This assertion is supported by the document "Re: Finding the caller in java," which indicates that "[y]ou can't find out which Object called a certain method unless you use the debugging API." The Examiner

notes that the specification as filed includes no discussion of using the debugging API to identify
an object.

Analysis of this factor in light of all the evidence of record therefore suggests that the
amount of experimentation required to make and use the invention is undue.

As to the existence of working examples, the specification does not describe a working
example. Analysis of this factor in light of all the evidence of record therefore suggests that the
amount of experimentation required to make and use the invention is undue.

As to the quantity of experimentation needed, there is no evidence in the record to
indicate the quantity of experimentation that one of ordinary skill in the art would need to
implement the present invention. Analysis of this factor in light of all the evidence of record
therefore does not suggest whether the amount of experimentation required to make and use the
invention is undue or not undue.

The majority of factors for which there is evidence suggest that undue experimentation is
required. See the discussion of the breadth of the claims, the state of the prior art, and the amount
of direction provided by the inventor. The level of predictability in the art suggests that undue
experimentation is not required. As to the other factors, the evidence of record is insufficient to
establish whether the amount of experimentation is undue or not undue. See the discussion of the
nature of the invention, the level of ordinary skill in the art, and the quantity of experimentation
needed. After weighing all of the factors and all the evidence of record, **the totality of the**

evidence suggests that it would require undue experimentation to make and use the

claimed invention.

Furthermore, although a patent need not teach, and preferably omits, what is well known

in the art (*In re Buchner*, 929 F.2d 660, 661, 18 USPQ2d 1331, 1332 (Fed. Cir. 1991);

Hybritech, Inc. v. Monoclonal Antibodies, Inc., 802 F.2d 1367, 1384, 231 USPQ 81, 94 (Fed.

Cir. 1986), *cert. denied*, 480 U.S. 947 (1987); and *Lindemann Maschinenfabrik GMBH v.*

American Hoist & Derrick Co., 730 F.2d 1452, 1463, 221 USPQ 481, 489 (Fed. Cir. 1984)), the

corollary to this statement is that a patent should disclose more detail concerning the features that

distinguish the claimed invention from the prior art. In this regard, the specification's lack of

direction, which is discussed above, appears especially critical.

35 USC § 112, Second Paragraph

Claims 6-9, 15-18, and 21 are rejected under 35 U.S.C. 112, second paragraph, as being

indefinite for failing to particularly point out and distinctly claim the subject matter which

applicant regards as the invention.

Each of the claims recites a Simple Object Access Protocol (or its abbreviation, SOAP),

but it is unclear to which Simple Object Access Protocol the claims are directed. Applicant's

specification incorporates by reference U.S. Patent No. 6,457,066 (hereinafter, "the '066

patent"), but also refers to the Apache Axis API (see paragraphs [0003]-[0004]). The Examiner

notes that the protocol described in the '066 patent is not the same as that implemented by

Apache Axis. The protocol described in the '066 patent is for accessing Microsoft COM

Automation objects using MIME-encoded messages (see col. 3, lines 1-64), while the protocol

that is implemented by Apache Axis is for accessing W3C SOAP web services using XML

messages (see "Introduction" on the Apache Axis Home Page). The protocols are similar in that

they are layered on top of HTTP and are used to access remote objects, but are otherwise entirely

different.

(10) Response to Argument

The Examiner will respond to arguments in the order in which they were presented in the

Brief.

Rejection of Claims 1-22 under 35 USC 112, First Paragraph

Appellant argues that the Examiner has required that the specification provide significant

guidance and direction despite the fact that the level of predictability in the computer arts is high.

The Examiner submits that this is but one factor to be considered when determining whether

there is sufficient evidence to support a determination that a disclosure does not satisfy the

enablement requirement and whether any necessary experimentation is "undue." After weighing

all of the factors and *all* the evidence of record, the *totality of the evidence* suggests that it would

require undue experimentation to make and use the claimed invention.

Appellant argues that "[p]aragraphs [0015]-[0022] and Figs. 2-5 of the present

application disclose a method for carrying out the claimed invention." The examiner submits

that, for the reasons given above in section (9), these paragraphs and figures are insufficient to

enable one of ordinary skill in the art to make and use the claimed invention without undue
experimentation.

Appellant argues that paragraph [0019], which states "[t]he identifier (fully qualified
class name) of the source of the SOAP call is stored in a SOAP header which is part of the
message transmitted from the client computer 11 to the data server 140 (via client server 120),"
describes how a comparison algorithm on a client server is used to identify an object on the
client computer that is invoking the object on a data server. The Examiner disagrees.

First, the paragraph omits *where* this header is inserted, and appears to be merely a
summation of what is occurring on the *client server*. In other words, this paragraph describes that
the *client server* determines the class name (by means unknown) and forwards it to the data
server to be reviewed by an administrator. This interpretation is consistent with the rest of the
specification, most notably paragraph [0015]. In contrast, Appellant's argument that this
paragraph describes the *client computer* sending a portion of its execution stack is *not* consistent
with the rest of the specification.

Second, this information in the header, regardless of which device sends it, only describes
a *class*, not an *object* as required by the claims. There is absolutely no mention in the
specification of using an algorithm to identify an object from a class name.

Third, if this header already contains the class name, it is unclear why the client server
would need a "comparison algorithm" to identify it. There is absolutely no mention in the
specification of using a comparison algorithm to analyze a message header.

Appellant argues that "[p]aragraph [0016] specifically recites an algorithm for implementing the claimed invention." The Examiner submits that for the reasons given above in section (9), this algorithm is insufficient to enable one of ordinary skill in the art to make and use the claimed invention without undue experimentation.

Appellant first contends that (1) "[t]his code is executed on the client computer." This contention is clearly inconsistent with both the specification, which indicates that the algorithm is operable on client server 120 (see paragraph [0016]), and with previous arguments made by the appellant. In particular, see the first full paragraph on page 8 of the reply filed November 26, 2007, which indicates that "[p]aragraph [0016] specifically recites an algorithm ***operable on the client server*** for implementing the claimed invention [emphasis in original]."

Appellant next contends that "a 'client server' need not be a separate machine" and cites the "Corba Glossary" as evidence. First, the Examiner submits that the instant invention is not directed toward a CORBA-based system, but toward a SOAP-based system. Second, the glossary merely describes a "server," not a "client server" as claimed and as used consistently throughout the specification. The Examiner submits that the term "client server" is not one which is commonly used in the art, and thus a proper construction would rely on the specification for guidance. Paragraph [0012] describes that "client computer 110 is networked with client server 120 via network link 150, such as a LAN or WAN connection (e.g., an Ethernet link)." LAN or WAN connections are not typically used to enable communication between programs on the same computer. Furthermore, Fig. 1 clearly illustrates client computer 110 and client server 120 as separate machines. Indeed, nothing in the specification or any document filed before the Brief suggests that client computer 110 and client server 120 are programs which run on a single

machine. Finally, if client computer 110 and client server 120 are programs running on the same system, it is unclear why the client computer would need to send a portion of its execution stack in the header of a SOAP message, as previously argued.

Rejection of Claims 6-9, 15-18, and 21 under 35 USC 112, Second Paragraph

Appellant argues that the term SOAP as used in the claims has a clear meaning, and that "[t]he protocol described in US Patent No. 6,457,066 ('the '066 patent') and the protocol implemented by Apache Axis were included as examples of implementations of SOAP," not as definitions of SOAP. However, the Examiner respectfully submits that no reasonably precise definition of SOAP exists which encompasses both the protocol described in the '066 patent and the protocol implemented by Apache Axis. The protocols are vaguely related in that they are used to access remote objects, but are wholly and fundamentally different in operation. See, for example, the evidence appendix at the end of this examiner's answer that includes the attached document "Scripting News: 9/27/2002," which discusses the '066 patent and indicates that it is not directed toward the XML-based SOAP described elsewhere in Appellant's specification. The examiner would note that this evidence is being supplied in response to newly raised arguments in the Appellant's appeal brief.

In effect, Appellant appears to be seeking protection on any protocol which happens to be named "SOAP" or "Simple Object Access Protocol." Although the presence of a trade name in a claim is not, *per se*, improper under 35 USC 112, its meaning must be established by an accompanying definition which is sufficiently precise and definite to be made a part of a claim.

See MPEP § 2173.05(u) and § 608.01(v). The Examiner submits that no such definition is currently of record.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/C. B./
Christopher Biagini
Examiner, Art Unit 2142

/Andrew Caldwell/
Supervisory Patent Examiner, Art Unit 2142

Conferees:

/Andrew Caldwell/
Supervisory Patent Examiner, Art Unit 2142

/Jason D Cardone/
Supervisory Patent Examiner, Art Unit 2145

EVIDENCE APPENDIX

The SOAP protocol described in Mein et al. (US Patent No. 6,457,066, September 2002) is not

the same as the XML-based protocol described elsewhere in Appellant's specification. See page

2 of "Scripting News: 9/27/2002" (http://www.scripting.com/2002/09/27.html), last viewed April

25, 2008.

Top > Dave's World > Weblog Archive > 2002 > September > 27   Previous/N

▤ **Friday, September 27, 2002**

Ed Cone: "As the only opponent of the Berman-Coble P2P
bill to speak at yesterday's hearings, Gigi Sohn got a
grilling from Howard Berman. 'He took every pot shot at
me he could,' says Sohn, president of an advocacy group
called Public Knowledge." ⸫

On BlogTree, Rebecca's Pocket lists Scripting News as a
parent weblog. Cool. ⸫

Le weblog de Jean-Yves sur Radio UserLand pour les
utilisateurs francophones. *Le rock star.* ⸫

Thanks to Charles Cooper for the good vibe. News.Com
still delivers great flow. ⸫

News.Com: "These versions of wireless networks using the
Wi-Fi, or 802.11b, standard create a wireless zone of up to
12 miles long, far beyond the usual 300-foot-radius range
that Wi-Fi typically achieves, Zakin said." ⸫

Jeremy Zawodny: Google, News and Making Money. ⸫

BBC "Until a year ago, customers at the chain were able
to download music and burn it onto CDs in-store." ⸫

Russ Lipton is looking for inspiration on his Radio book ⸫

Last update: Friday, Septembe
27, 2002 at 5:39 PM Eastern.

Dave Winer 🖵
Search   Go

OPML ™

MORNING
COFFEE
NOTES

September 2002
Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	**27**	28
29	30					

Aug Oct

Lisa Rein: Give Peace a Chance. *

Did Microsoft patent SOAP? *

This newly issued patent (9/24/02) makes it appear that
they did. However this is not the SOAP that's in use today.
One clue is the date it was filed, 11/10/97. Work on
XML-based SOAP didn't begin until March 1998. Further,
the description is of something quite different from what
we call SOAP today. People at Microsoft liked the name
SOAP, and when the binary transport for COM was
stillborn, they wanted to re-use it for the XML-based
SOAP. I confirmed with Microsoft that they had not
patented XML-based SOAP, and they said they hadn't.
Another large software company told me at the time that
they were sure that they had. No matter, had Microsoft
wanted to patent XML-based SOAP they would have
needed to get me on board, and I never gave permission to
do that.

*Postscript: I have confirmation from Microsoft people in
the know that this interpretation of the patent is correct.
We may have some more info tomorrow.*

What about patents? *

However, at lunch today with an old friend, we talked
about new ideas for spreadsheets, and I said if I worked on
that, I definitely *would* file for patents. After watching so
many pigs feed at the trough, I realized that being the only
honorable person is totally unfair to me. Further, to other
anti-patent people, generosity seems to buy no
consideration. If I have some patents, they'll have to
negotiate. So if I invested the time to create a better
spreadsheet (just an example), I would patent it, and make
my competitors pay for the right to use my ideas. Maybe
I'll change my mind again, it's quite possible, and it's also
possible that I'll never have a unique software idea again,
so this might be moot. If you're anti-software-patents, give
it some thought. You might be being a chump too.

Change in RSS 2.0 support in Radio *

This morning an esoteric update to the RSS feed generator
in Radio. We now omit the xmlns attribute on the rss
element because some parsers, especially homegrown

parsers, can't correctly interpret it.

This issue arose when Ovidu Predescu had errors polling Sam Ruby's and Simon Fell's feeds. Anyway, the fix was to drop the xmlns attribute. It's still RSS 2.0 without it, and Radio wasn't actually using any namespaces, so there's no functionality change, and it should unbreak Ovidu's parser, and any others that have trouble with namespaces.

Now, I'm not backing off namespaces in the Scripting News feed, through its use of the blogChannel module. This way, any breakage that's reported will come just for my weblog, not for Radio users' weblogs.

I apologize for the difficulties. I promise, it's for a good cause -- if we wanted to allow modularity (we do) in the XML feeds we would have hit this problem at some point. Now if people are concerned, they can update right now and all aggregators should be happy and life goes on.

Previous/Next